



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/686,746	10/16/2003	Takeshi Tanaka	82478-1300	8702

21611 7590 02/02/2007  
SNELL & WILMER LLP  
600 ANTON BOULEVARD  
SUITE 1400  
COSTA MESA, CA 92626

EXAMINER
----------

LAI, VINCENT

ART UNIT	PAPER NUMBER
----------	--------------

2181

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
3 MONTHS	02/02/2007	PAPER

**Please find below and/or attached an Office communication concerning this application or proceeding.**

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

<b>Office Action Summary</b>	<b>Application No.</b> 10/686,746	<b>Applicant(s)</b> TANAKA ET AL.	
	<b>Examiner</b> Vincent Lai	<b>Art Unit</b> 2181	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 13 November 2006.
- 2a) ☐ This action is FINAL.                      2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 1,3-10 and 14 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1,3-10 and 14 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All    b) ☐ Some \*    c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

- |  |   |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)          | 4) <input type="checkbox"/> Interview Summary (PTO-413)           |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date: _____                                      |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)          | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| Paper No(s)/Mail Date: _____   | 6) <input type="checkbox"/> Other: _____                          |

## **DETAILED ACTION**

### ***Response to Amendment***

1. Acknowledgment is made of amendments made to the claims.

### ***Continued Examination Under 37 CFR 1.114***

2. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 19 October 2006 has been entered.

### ***Priority***

3. Acknowledgment is made of applicant's claim for foreign priority under 35 U.S.C. 119(a)-(d).

### ***Response to Arguments***

4. Applicant's arguments with respect to claims 1, 3-10, and 14 have been considered but are moot in view of the new ground(s) of rejection.

### ***Claim Objections***

Art Unit: 2181

5. Claim 4 is objected to because of the following informalities: The claim recited limitations of "a register that includes a plurality of register files," which is seemingly contradictory of what a register file is. One having ordinary skill in the art would recognize that a register file is an array of registers and thus it would not make sense for a register to include a plurality of arrays of registers. Examiner recognizes that Applicant may be its own lexicographer and thus requests Applicant to further detail how such contradictions may be resolved.

6. Claim 8 is objected to because of the following informalities: It appears that the word "fetching" has been spelled "feting" in the newly amended portion of claim 8. A similar amendment was made in claim 9, in which "fetching" was spelled correctly.

Appropriate correction is required.

***Claim Rejections - 35 USC § 102***

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

7. Claims 1, 3-10, and 14 are rejected under 35 U.S.C. 102(b) as being anticipated by Eickenmeyer et al (U.S. Patent # 5,355,460), herein referred to as Eickenmeyer.

Art Unit: 2181

As per **claim 1**, Eickenmeyer discloses a parallel execution processor comprising:

a plurality of processing elements (See figure 1: Function Units 13-15);

an obtaining unit operable to obtain an instruction sequence including one or more instructions (See figure 1 and column 6, lines 11-17: The compound instruction cache 12 hold sequences of instructions);

a decoding unit operable to decode the obtained instruction sequence into the one or more instructions (See figure 8 and column 13, lines 53-64: Decoders 40, 41, and 45 are capable of sorting out the various op code to group instructions together);

a group forming unit (See figure 1: Instruction Compounding Unit 11) operable to form the processing elements into as many groups as the number of instructions included in the instruction sequence (See column 9, lines 44-49: The ideal case is when there is as many processing elements as instruction sequences); and

an execution controlling unit operable to assign the one or more instructions decoded by the decoding unit to the groups of the processing elements (See figure 8 and column 13, lines 53-64: Decoders 40, 41, and 45 are capable of sorting out the various op code to group instructions together), so that one group of processing elements received one instruction, and control the processing elements so that (i) the instructions received by the groups are executed in parallel (See column 5, lines 30-35: The instructions are grouped in such a manner such that they can be run in parallel), and (ii) in each group, all processing elements in the group are employed in parallel for the

Art Unit: 2181

execution of the received instruction (See column 5, lines 60-64: The functional units run in parallel).

**Claim 2** has been cancelled.

As per **claim 3**, Eickenmeyer discloses the parallel execution processor of claim 1, wherein

when the number of instructions included in the instruction sequence is one, the group forming unit forms all of the processing elements into one group (See column 9, lines 44-49: This is the case when  $N = 1$ ), and

when the number of instructions included in the instruction sequence is two, the group forming unit forms all of the processing elements into two groups so that the two groups contain an equal number of processing elements (See column 9, lines 44-49: This is the case when  $N = 2$ ).

As per **claim 4**, Eickenmeyer discloses the parallel execution processor of claim 3, further comprising

a register that includes a plurality of register files (See figure 6: Tagged instruction register 27) each of which corresponds to a different one of the processing elements (See column 11, lines 32-38: Registers are tagged for each instruction group), wherein

the instruction sequence includes a first instruction and a second instruction (See column 9, lines 44-49: This is the case when  $N = 2$ ),

the register files are arranged in the register so that first-group register files and second-group register files alternate (See figure 6 and 9 and column 9, lines 44-49: This would be the case where  $N = 2$  and the choice would be either tag0 or tag1), (i) the first-group register files each storing therein a piece of data to be processed when the first instruction is executed (See figure 6 and 9 and column 9, lines 44-49, and column 11, lines 32-38: This would be the case where  $N = 2$  and the choice would be either tag0 or tag1. Instructions associated with tag0 will be stored in corresponding registers) and (ii) the second-group register files each storing therein a piece of data to be processed when the second instruction is executed (See figure 6 and 9, column 9, lines 44-49: This would be the case where  $N = 2$  and the choice would be either tag0 or tag1. Instructions associated with tag1 will be stored in corresponding registers),

when the number instructions included in the instruction sequence is two, the group forming unit forms the processing elements corresponding to the first-group register files into one of the two groups, and the processing elements corresponding to the second-group register files into the other group (See column 9, lines 44-49: This is the case when  $N = 2$ ), and

each of the processing elements obtains the piece of data to be processed from the corresponding register file (See figure 1 and column 6, lines 11-17: Only the corresponding functional unit will process the group data in the registers).

Art Unit: 2181

As per **claim 5**, Eickenmeyer discloses the parallel execution processor of claim 4, wherein

the register files are formed into a plurality of pairs (See column 10, lines 17-19: Pairwise compounding keeps ensures pairs of register files), keeping an order in which the register files are arranged in the register (See column 11, lines 32-38: Tags are used to ensure that register files are kept in order),

each of the instructions includes a piece of selection information indicating which piece of data each processing element should obtain (See figure 6: Each register file has instructions), selecting out of (a) the piece of data stored in the corresponding register file and (b) the piece of data stored in a register file with which the corresponding register file is paired (See column 12, lines 22-44: Data can be grabbed from another group if needed but instruction will be run if no dependencies are outstanding), and

each of the processing elements obtains the piece of data to be processed from the register file indicated in each piece of selection information (See figure 1 and column 6, lines 11-17: Only the corresponding functional unit will process the group data in the registers).

As per **claim 6**, Eickenmeyer discloses the parallel execution processor of claim 3, wherein

when the number of instructions included in the instruction sequence is two, the execution controlling unit includes:



Art Unit: 2181

a storing unit (See column 13, lines 16-20: The compound analyzer 22 stores the combination options) that stores therein a plurality of combination options based on which of the processing elements should belong to each of the two groups (See column 13, lines 23-31: Examples of combination rules), the combination options being prepared for each of a plurality of grouping procedures;

a grouping information obtaining unit (See figure 1: Instruction Compounding Unit 11) operable to obtain a piece of grouping information indicating which one of the grouping procedures should be used (See column 11, lines 32-38: Tags are used to identify groups); and

a selecting unit operable to select one of the combination options according to the obtained piece of grouping information (See figure 1 and column 6, lines 11-17: Only the corresponding functional unit will process the group data in the registers).

As per **claim 7**, Eickenmeyer discloses the parallel execution processor of claim 3, wherein

when the number of instructions included in the instruction sequence is two, the execution controlling unit includes:

a grouping information obtaining unit (See figure 1: Instruction Compounding Unit 11) operable to obtain a piece of grouping information indicating to which one of the two groups, each of the processing elements should belong (See column 11, lines 32-38: Tags are used to identify groups); and

Art Unit: 2181

a grouping unit operable to form the processing elements into the two groups according to the obtained piece of grouping information (See column 9, lines 44-49: This is the case when  $N = 2$ ).

As per **claim 8**, Eickenmeyer discloses the parallel execution processor of claim 1, further comprising

a fetching unit (See figure 10: Fetch/Issue Control Unit) operable to fetch a piece of data which is of a predetermined length and has a format field and a data field (See column 15, line 68- column 16, line 11: The fetch unit utilizes op code, which has a predetermined length, format and data field, to fetch data), wherein

each of the instructions includes an OP code and an operand (See column 15, line 68- column 16, line 11: Op code is used),

a positioning pattern is written in the format field (See figure 7 and column 11, lines 32-38: Tags are used to identify instructions, which are held in the instruction register 21), the positioning pattern being for positioning OP codes and operands in the data field (See column 11, lines 32-38: Tags are used to identify groups and used to place instructions into the corresponding groups),

in the piece of data, one or more OP codes and one or more operands are arranged in the data field in an order defined by the positioning pattern written in the format field (See figure 7: The instruction register 21 can hold more than one instruction and the system is able to identify the start on one by its tag),

the obtaining unit obtains, as the instruction sequence, the piece of data of the predetermined length fetched by the fetching unit [sic] (See figure 1 and column 6, lines 11-17: The compound instruction cache 12 fetches and holds sequences of instructions),

the decoding unit extracts, from the piece of data, the one or more OP codes and the one or more operands, according to the positioning pattern so as to decode the OP codes and the operands of the instructions (See figure 8 and column 13, lines 53-64: Decoders 40, 41, and 45 are capable of sorting out the various op code to group instructions together), and

the execution controlling unit assigns, in the defined order, the decoded instructions to the groups (See column 5, lines 30-35: The instructions are grouped by encoding in their tags).

As per **claim 9**, Eickenmeyer discloses the parallel execution processor of claim 1, further comprising:

a fetching unit (See figure 10: Fetch/Issue Control Unit) operable to fetch a piece of data which is of a predetermined length (See column 15, line 68- column 16, line 11: The fetch unit utilizes op code, which has a predetermined length); and

a storing unit operable to store therein a predetermined positioning pattern for OP codes and operands (See figure 1: Compound Instruction cache 12), wherein

each of the instructions includes an OP code and an operand (See column 15, line 68- column 16, line 11: Op code is used),

Art Unit: 2181

one or more OP codes and one or more operands are arranged in the piece of data in an order defined by the predetermined positioning pattern (See column 11, lines 32-38: Tags are used to identify groups and used to place instructions into the corresponding groups),

the obtaining unit obtains, as the instruction sequence, the piece of data of the predetermined length fetched by the fetching unit (See figure 1 and column 6, lines 11-17: The compound instruction cache 12 fetches and holds sequences of instructions),

the decoding unit extracts, from the piece of data, the one or more OP codes and the one or more operands, according to the positioning pattern stored in the storing unit so as to decode the OP codes and the operands of the instructions (See figure 8 and column 13, lines 53-64: Decoders 40, 41, and 45 are capable of sorting out the various op code to group instructions together), and

the execution controlling unit assigns, in the defined order, the decoded instructions to the groups (See column 5, lines 30-35: The instructions are grouped by encoding in their tags).

As per **claim 10**, Eickenmeyer discloses the parallel execution processor of claim 1, wherein

when the instruction sequence obtained by the obtaining unit includes two or more instructions and one of the instructions instructs that processing elements included in some of the groups should halt operation, the execution controlling unit controls the processing elements included in those groups so that those processing

Art Unit: 2181

elements halt operation (See column 12; lines 22-44: A halt is necessary when a dependency is found, which would halt the group).

**Claims 11-13** have been cancelled.

**Claim 14** is rejected for reasons similar to that of claim 1. Claim 14 is the method of the parallel execution processor of claim 1.

**Claim 15** has been cancelled.

### ***Conclusion***

8. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Vincent Lai whose telephone number is (571) 272-6749. The examiner can normally be reached on M-F 8:00-5:30 (First BiWeek Friday Off).

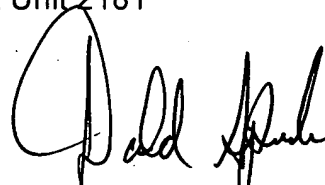
If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Donald Sparks can be reached on (571) 272-4201. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Art Unit: 2181

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

vi  
January 24, 2007

Vincent Lai  
Examiner  
Art Unit 2181

A handwritten signature in black ink, appearing to read "Donald Sparks", is written over a rectangular stamp area.

**DONALD SPARKS**  
**SUPERVISORY PATENT EXAMINER**